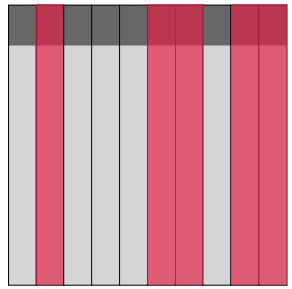
Principal Component Analysis

Lecture 13

Termeh Shafie

1

Dimensionality Reduction

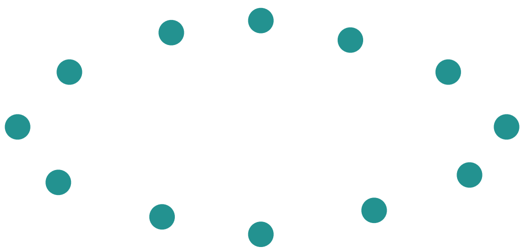


Principal Component Analysis

- does not drop variables
- creates new variables to describe the information in our data, so called **principal components**

2

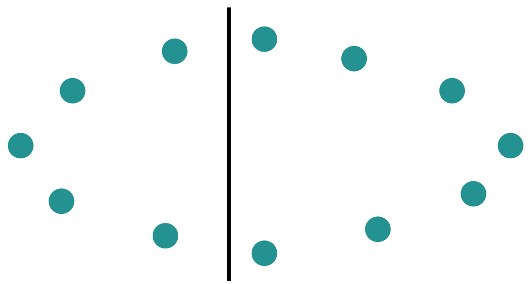
What is PCA?



which is the direction where the variation the largest?

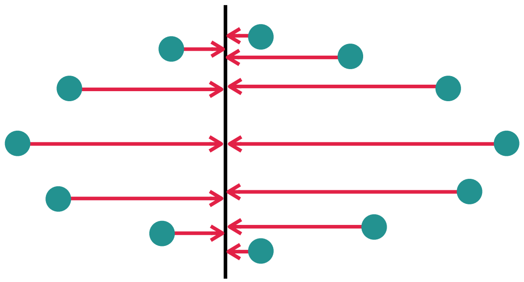
3

What is PCA?



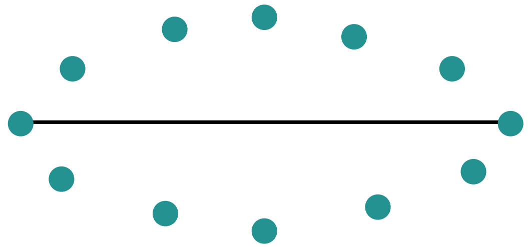
4

What is PCA?



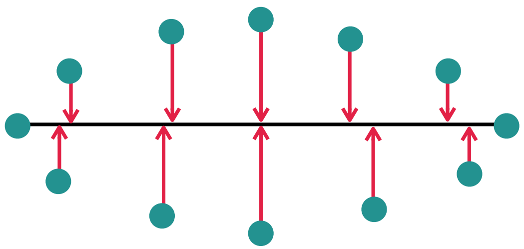
5

What is PCA?



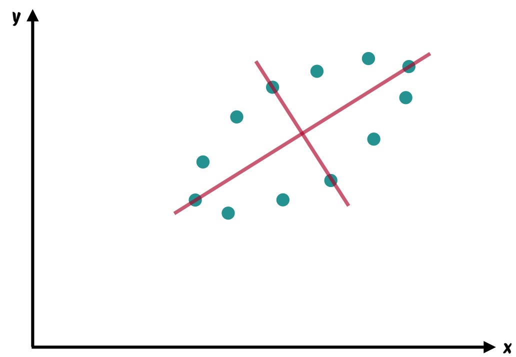
6

What is PCA?



7

What is PCA?

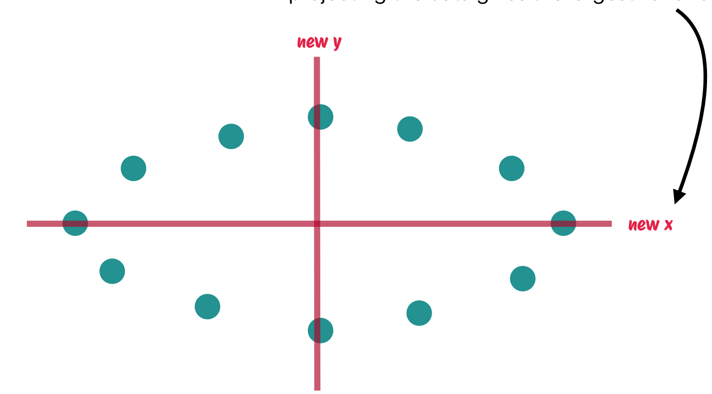


⟹ PCA aims to find a new coordinate system for your data

8

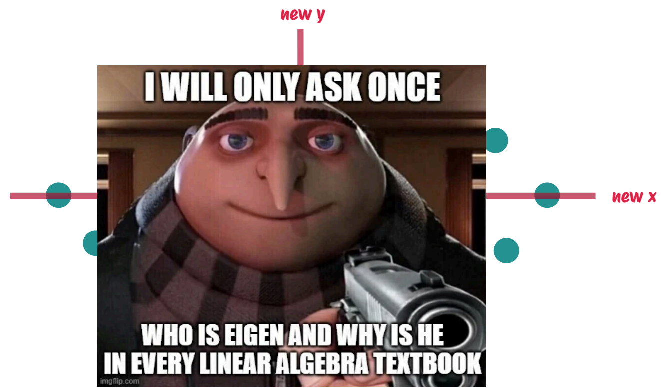
What is PCA?

The first principal component is the direction onto which projecting the data gives the largest variance.

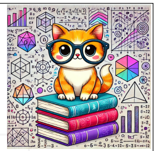


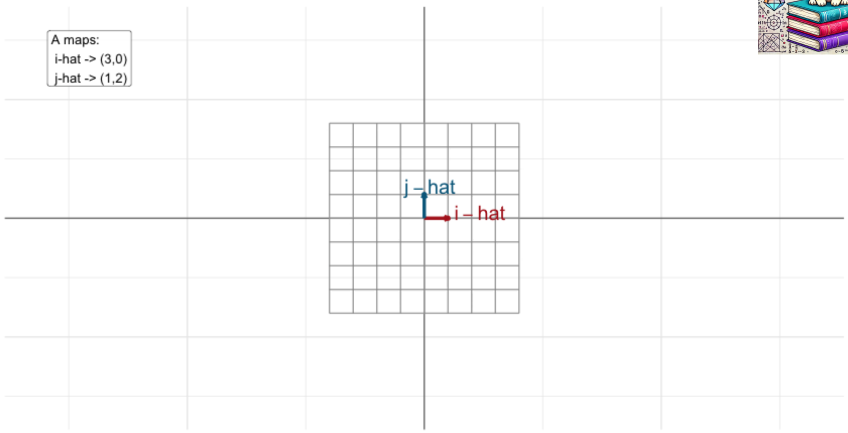
⟹ PCA aims to find a new coordinate system for your data

What is PCA?

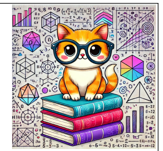


Eigendecomposition



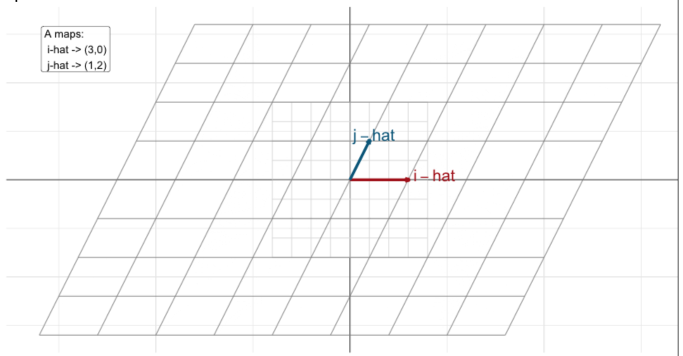


Eigendecomposition

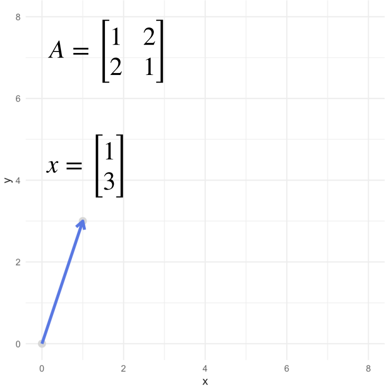


most vectors get knocked of their span

but some stay put and only get stretched/squished/reversed

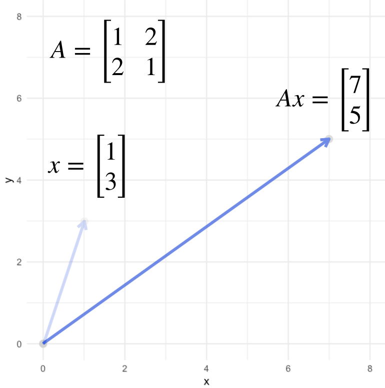


Eigendecomposition



What happens when a matrix hits a vector?

Eigendecomposition

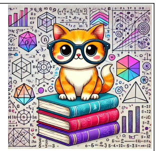


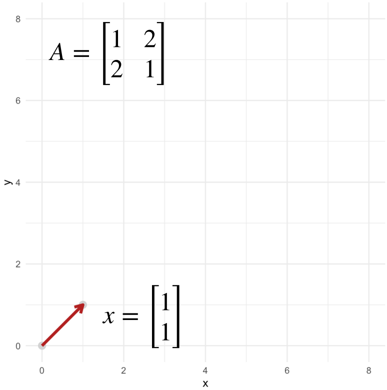
What happens when a matrix hits a vector?

The vector transforms into a new vector

- it strays from its path
- it may get scaled: stretched (longer) or squished (shorter)

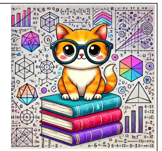
Eigendecomposition

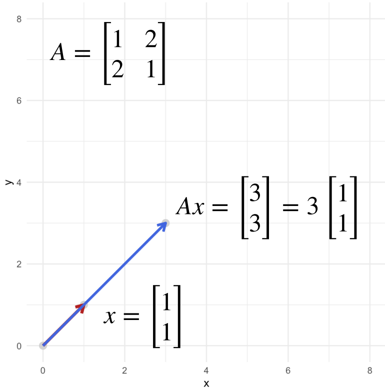




For a given square matrix A, there are special vectors which refuse to stray from their path

Eigendecomposition

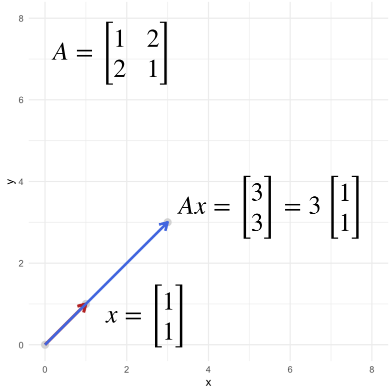




For a given square matrix A, there are special vectors which refuse to stray from their path

These vectors are called eigenvectors

Eigendecomposition



For a given square matrix A, there are **special vectors** which refuse to stray from their path

These vectors are called **eigenvectors**

Formally, $Ax = \lambda x$
where λ are the eigenvalues determining the scale, but directions remains the same (x)

Several properties of matrices can be analyzed based on their eigenvalues.

Eigendecomposition

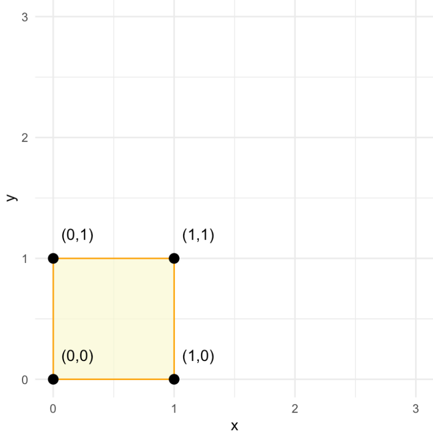
The eigenvectors of a square matrix A having distinct eigenvalues are linearly independent.

The eigenvectors of a square **symmetric** matrix are **orthogonal**.

The eigenvectors of a square symmetric matrix can thus form a convenient basis.

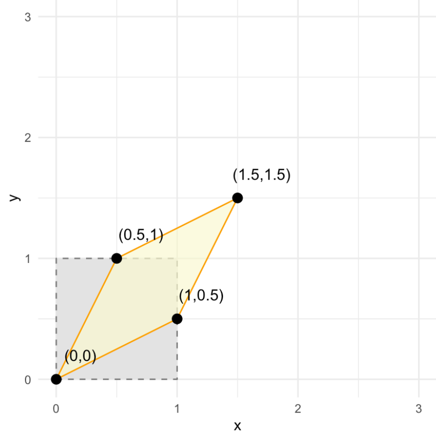
$$\mathrm{Cov}(\mathbf{x}) = \begin{bmatrix} \mathrm{Var}(x_1) & \mathrm{Cov}(x_1,x_2) & \mathrm{Cov}(x_1,x_3) & \cdots & \mathrm{Cov}(x_1,x_n) \\ \mathrm{Cov}(x_2,x_1) & \mathrm{Var}(x_2) & \mathrm{Cov}(x_2,x_3) & \cdots & \mathrm{Cov}(x_2,x_n) \\ \mathrm{Cov}(x_3,x_1) & \mathrm{Cov}(x_3,x_2) & \mathrm{Var}(x_3) & \cdots & \mathrm{Cov}(x_3,x_n) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(x_n,x_1) & \mathrm{Cov}(x_n,x_2) & \mathrm{Cov}(x_n,x_3) & \cdots & \mathrm{Var}(x_n) \end{bmatrix}$$

Eigendecomposition



Eigendecomposition

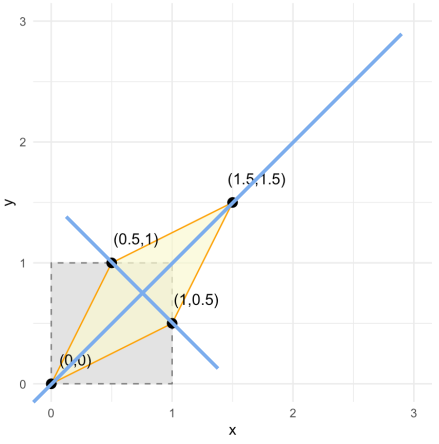
$\times \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$



Eigendecomposition

$$\times \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

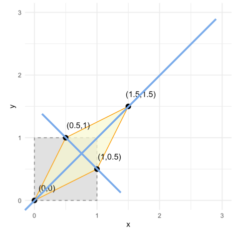
'stretch' and 'squish'
direction? (eigenvectors)
how much? (eigenvalues)



Eigendecomposition

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad Ax = \lambda x$$

to find the eigenvalues λ we can solve the so called **characteristic polynomial**



$$|A - \lambda I| = 0 \quad \text{where} \quad \lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$\begin{bmatrix} 1-\lambda & 0.5 \\ 0.5 & 1-\lambda \end{bmatrix}$$

$$|A - \lambda I| = (1-\lambda)(1-\lambda) - (0.5)(0.5)$$

$$= \lambda^2 - 2\lambda + 0.75$$

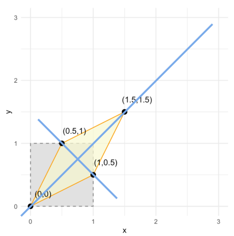
solve the roots to get **eigenvalues**: $(\lambda - 1.5)(\lambda - 0.5) \implies \lambda = [1.5,\ 0.5]$

Eigendecomposition

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \qquad Ax = \lambda x$$

plug eigenvalues back and get **eigenvectors** (direction)

$$\lambda = [1.5,\ 0.5] \rightarrow \begin{bmatrix} 1-\lambda & 0.5 \\ 0.5 & 1-\lambda \end{bmatrix}$$



$$\rightarrow \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix} \qquad \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}$$

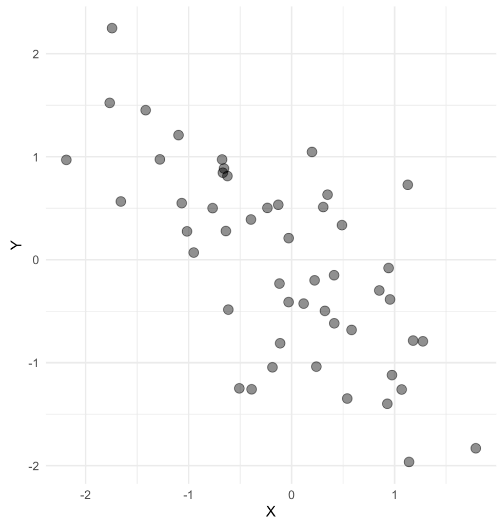
PCA Summary

1. Principal Component Analysis (PCA) finds a new set of orthogonal axes that best explain the variance in a dataset.
2. Given the covariance matrix Σ, PCA solves the optimization problem of finding directions that maximize the variance of the projected data. Maximizing this quantity leads to the eigenvalue problem $\Sigma \mathbf{w} = \lambda \mathbf{w}$, where the variance along a unit direction $\mathbf{w}$ is $\mathbf{w}^\top \Sigma \mathbf{w}$.
3. The principal components are the eigenvectors of the covariance matrix, ordered by decreasing eigenvalues. The eigenvalue gives the amount of variance explained by its corresponding eigenvector.
   - PC_1 is the eigenvector with largest eigenvalue and captures the maximum variance.
   - PC_2 is the eigenvector with second-largest eigenvalue and captures the maximum remaining variance subject to being orthogonal to PC_1.
4. Because the covariance matrix is symmetric, its eigenvectors are orthogonal, which ensures that principal components are uncorrelated.

Example

in PCA we perform eigendecomposition on the covariance matrix of the data

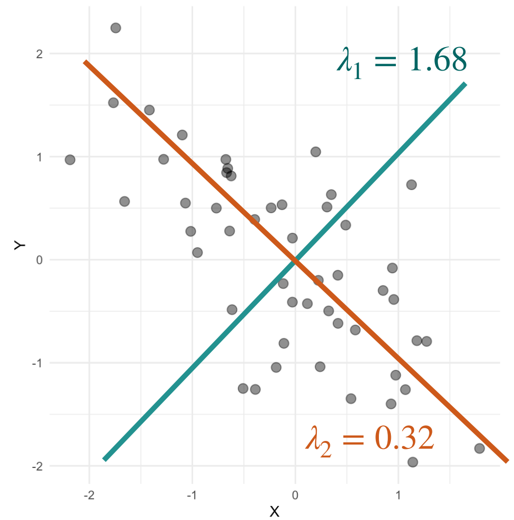
$$\mathrm{Cov}(\mathbf{x}) = \begin{bmatrix} 1 & -0.69 \\ -0.69 & 1 \end{bmatrix}$$



Example

in PCA we perform eigendecomposition on the covariance matrix of the data

$$\mathrm{Cov}(\mathbf{x}) = \begin{bmatrix} 1 & -0.69 \\ -0.69 & 1 \end{bmatrix}$$



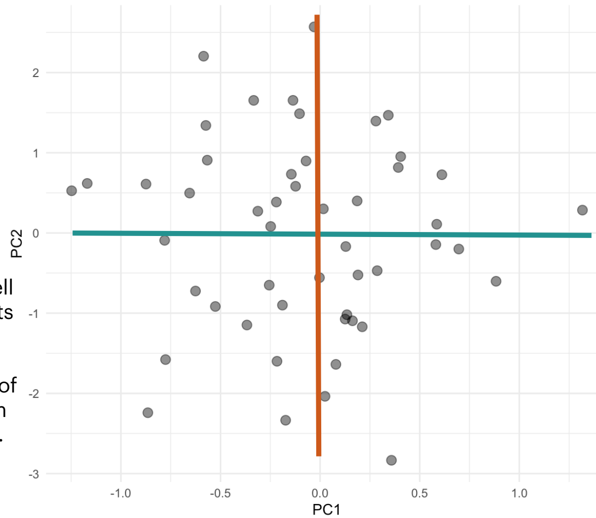
Example

$$\text{eigenvect}_1 = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}$$

$$\text{eigenvect}_2 = \begin{bmatrix} 0.71 \\ -0.71 \end{bmatrix}$$

the **loadings** (eigenvectors/weights) tell us how much of the original data points go into our new PC variables

the **scores** are the transformed values of the data in the new coordinate system defined by the principal components.



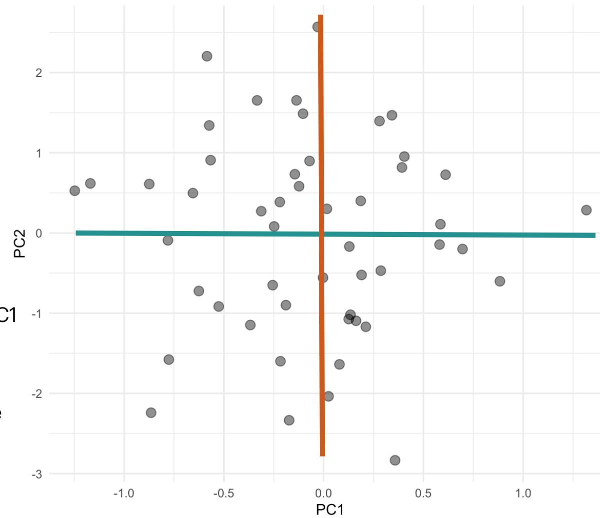
Example

$$\text{eigenvect}_1 = \begin{bmatrix} 0.71 \\ 0.71 \end{bmatrix}$$

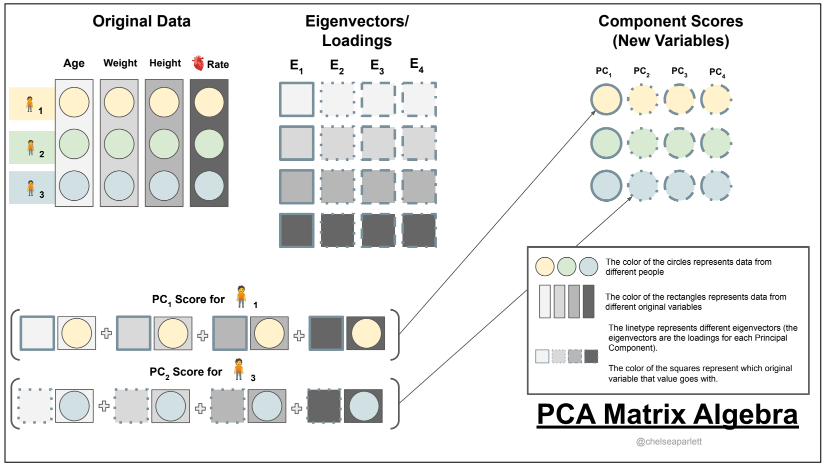
$$\text{eigenvect}_2 = \begin{bmatrix} 0.71 \\ -0.71 \end{bmatrix}$$

X and Y contribute equally (0.71), so PC1 represents an overall trend

X and Y contribute oppositely (0.71 and -0.71), so PC2 measures the difference between them



A PC **score** tells you how much of a principal component direction is present in a data point, computed by projecting the data onto that eigenvector.



Small Example

Let the two variables be x and y.

Observations:
A: (2, 1)
B: (4, 3)
C: (6, 5)

Mean-center the data
A: (-2, -2)
B: (0, 0)
C: (2, 2)

Using sample covariance

$$\Sigma = \begin{bmatrix} \mathrm{Var}(x) & \mathrm{Cov}(x,y) \\ \mathrm{Cov}(x,y) & \mathrm{Var}(y) \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$$

Eigenvectors of Σ:

PC_1 direction (largest eigenvalue):

$$\mathbf{v}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \approx \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$$

PC_2 direction (other eigenvector):

$$\mathbf{v}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \approx \begin{bmatrix} 0.707 \\ -0.707 \end{bmatrix}$$

$\Longrightarrow$ the loadings for PC_1 are 0.707 on x and 0.707 on y.

$\Longrightarrow$ the loadings for PC_2 are 0.707 on x and —0.707 on y.

Small Example

PC_1 score = $0.707 \cdot x_c + 0.707 \cdot y_c$

where (x_c, y_c) is a centered point

Calculate:

A: (-2,-2)

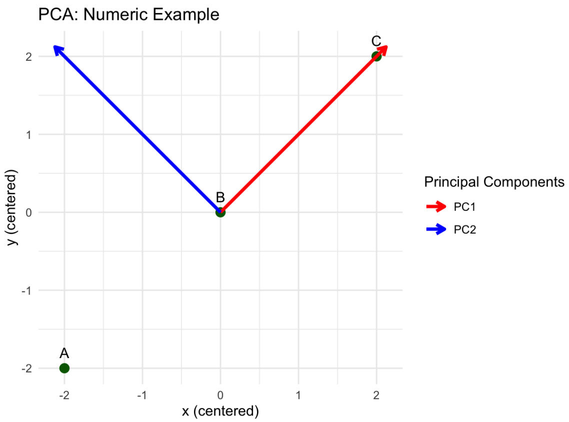
$\Longrightarrow$ 0.707(-2) + 0.707(-2) = -2.828

B: (0,0)

$\Longrightarrow$ 0

C: (2,2)

$\Longrightarrow$ 0.707(2) + 0.707(2) = 2.828



PC_2 scores are all 0 meaning geometrically, the data lies perfectly on a line, so there's no variance in the perpendicular direction (a zero eigenvalue case).

Example: Loadings

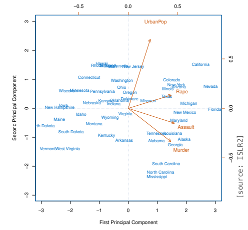
Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
Income	0.314	0.145	-0.676	-0.347	-0.241	0.494	0.018	-0.030
Education	0.237	0.444	-0.401	0.240	0.622	-0.357	0.103	0.057
Age	0.484	-0.135	-0.004	-0.212	-0.175	-0.487	-0.657	-0.052
Residence	0.466	-0.277	0.091	0.116	-0.035	-0.085	0.487	-0.662
Employ	0.459	-0.304	0.122	-0.017	-0.014	-0.023	0.368	0.739
Savings	0.404	0.219	0.366	0.436	0.143	0.568	-0.348	-0.017
Debt	-0.067	-0.585	-0.078	-0.281	0.681	0.245	-0.196	-0.075
Credit cards	-0.123	-0.452	-0.468	0.703	-0.195	-0.022	-0.158	0.058

Example: USA Arrests (ISLR)

- For each state in the US:
  - number of arrests per 100 000 residents for `Assault`, `Murder` and `Rape`.
- Included is also the percent of the population in each state living in urban areas
- PC score vectors have length $n = 50$
- PC loading vectors have length $p = 4$
- PCA performed after standardizing each variable

Example: USA Arrests Biplot

	PC1	PC2
`Murder`	0.5358995	−0.4181809
`Assault`	0.5831836	−0.1879856
`UrbanPop`	0.2781909	0.8728062
`Rape`	0.5434321	0.1673186



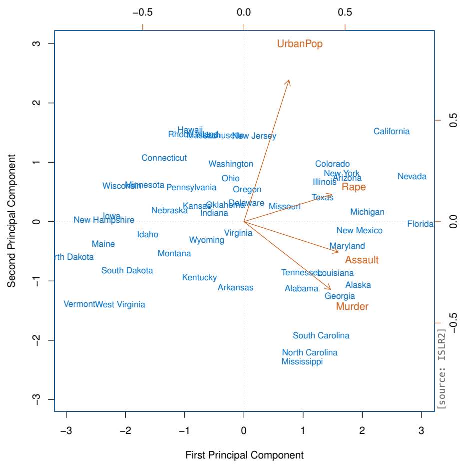
- **PC1**
  High loadings for Murder (0.536), Assault (0.583), and Rape (0.543):
  - These three variables contribute strongly and approximately equally to PC1.
  - PC1 could represent a general "crime severity" axis, as it captures patterns where these types of crimes tend to vary together.

  UrbanPop (0.279) has a smaller contribution:
  - Population density has less influence on PC1 compared to the crime-related variables.
- **PC2**
  High loading for UrbanPop (0.873):
  - PC2 is primarily influenced by UrbanPop.
  - This suggests PC2 captures variation in population density that is independent of crime severity.

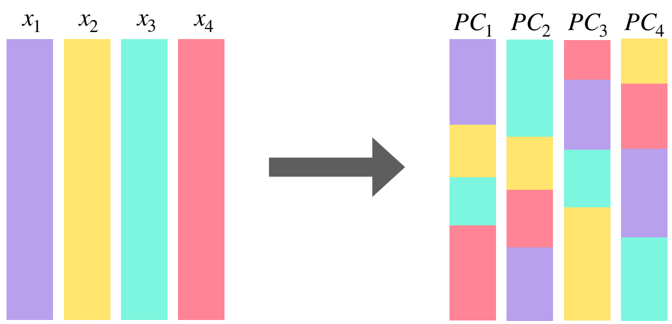
  Negative contributions from Murder (-0.418) and Assault (-0.188):
  - Murder and Assault negatively influence PC2, indicating areas with high UrbanPop might have slightly lower relative crime rates.

Example: USA Arrests Biplot

	PC1	PC2
`Murder`	0.5358995	−0.4181809
`Assault`	0.5831836	−0.1879856
`UrbanPop`	0.2781909	0.8728062
`Rape`	0.5434321	0.1673186

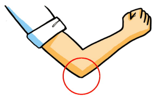


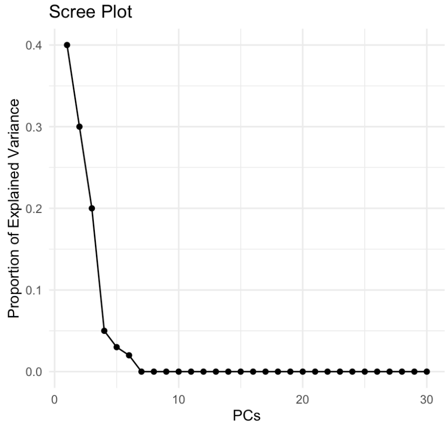
Dimensionality Reduction

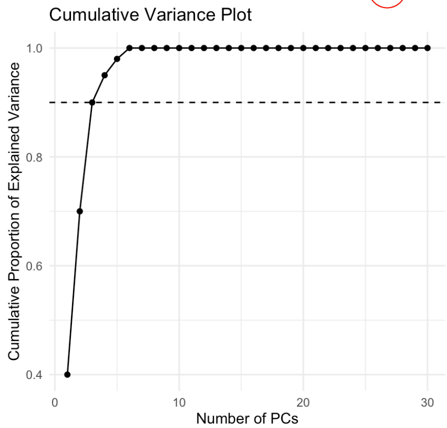


note: this is **not** variable selection

Scree and Cumulative Variance Plots

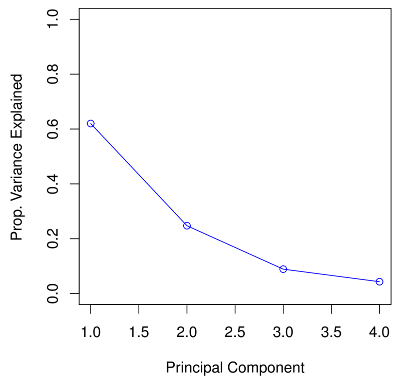


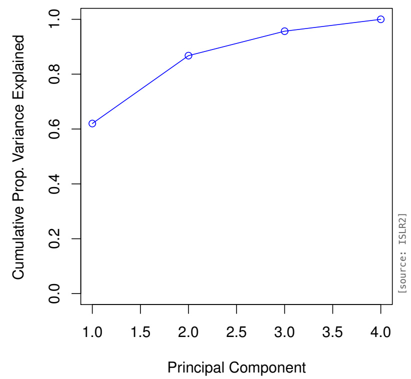




Scree and Cumulative Variance Plots

	PC1	PC2
Murder	0.5358995	−0.4181809
Assault	0.5831836	−0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186





Principal Components Regression (PCR)

1. Use PCA to find principal components among the covariates
2. Use these principal components as independent variables in a LS regression to get a vector of coefficient estimates
3. Transform this vector back to the scale of the actual covariates, using the selected PCA loadings
4. The final PCR estimator will have same dimension equal to the total number of covariates

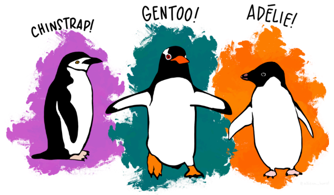
Single Value Decomposition

PCA can also be done using SVD on the data matrix instead

This Week's Practical

PCR and PCA

perform PCA on penguin body features



source: @allison_horst https://github.com/allisonhorst/penguins